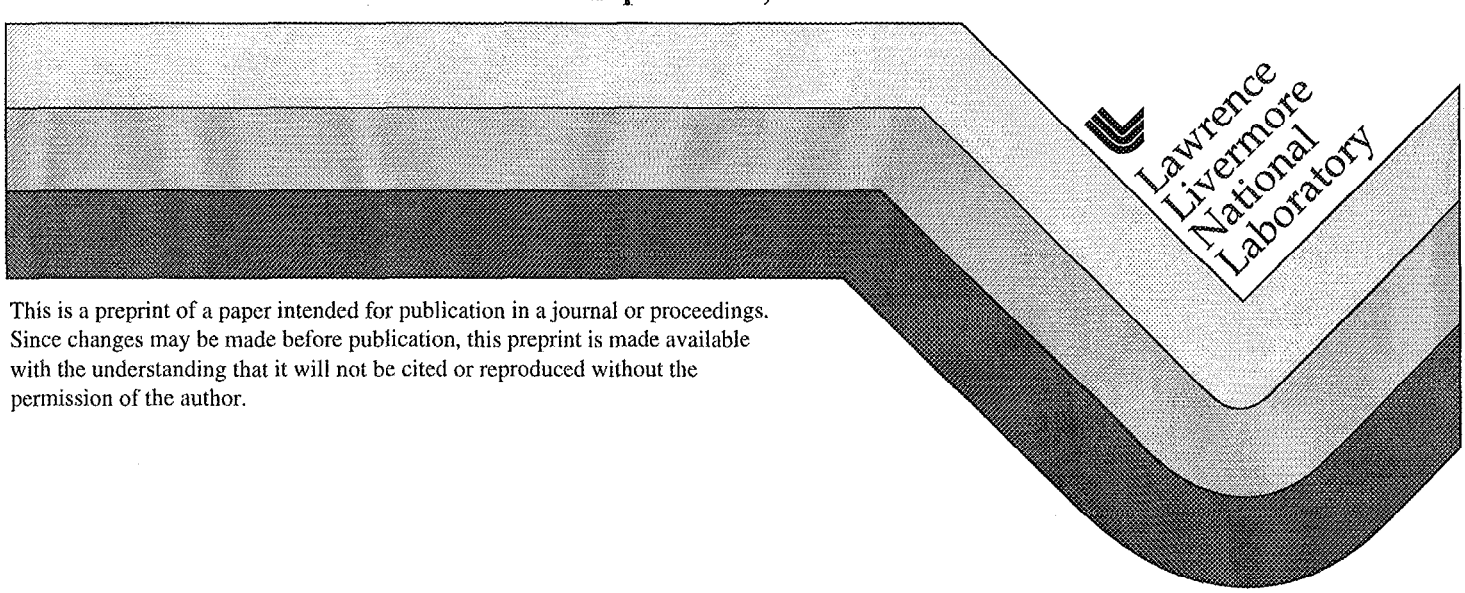
UCRL-JC-131828
PREPRINT

Neural Network Approximation of Numeric Subsurface Models in Combinatorial Optimization

V.M. Johnson
L.L. Rogers

This paper was prepared for submittal to the
International Joint Conference on Information Science
Research Triangle Park, NC
October 23-28, 1998

September 4, 1998



Neural Network Approximation of Numeric Subsurface Models in Combinatorial Optimization

Virginia M. Johnson and Leah L. Rogers

Lawrence Livermore National Laboratory
L-208, PO Box 808, Livermore, CA 94551, USA
vmjohnson@llnl.gov and rogers11@llnl.gov

1. INTRODUCTION

The ANN-GA approach to design optimization integrates two well-known computational technologies, artificial neural networks (ANNs) and the genetic algorithm (GA), with a simple scheme for exploiting a network of common workstations to reduce the computational burden associated with applying formal optimization techniques to subsurface engineering problems. The greatest computational investment in a design project of the kind which will be described in this paper is in the simulation of physical processes needed to calculate the cost function. The ANN-GA methodology addresses this problem by training ANNs to stand in for the simulator during the course of a search directed by the GA. The ANNs are trained and tested from examples stored in a re-usable knowledge base of representative simulations which relate variations in the design parameters to predicted outcomes for the particular engineering problem being studied. The maximum amount of information from each simulation is saved, subject to storage limitations. The creation of the knowledge base is itself a sizeable computational investment, one that pays off if it is used to train a variety of networks for different searches and/or for use in other contexts such as sensitivity analyses. A diagram of the components of the methodology is given in Figure 1. Applications of the methodology have been reported in [1-3].

From the standpoint of the computational time devoted to simulation, the ANN-GA methodology converts a serial search into a parallel search. The cost function information required during the usual GA-driven serial search is provided by neural networks trained from information in the initial knowledge base. The knowledge base is created from representative simulations which, because they are independent of each other, are run in parallel by distributing them over a network of workstations. The effectiveness of the approach is dependent on the representativeness of the sample of simulations constituting the initial knowledge base. The current challenge is to ensure that this representativeness is adequate for the problem being studied.

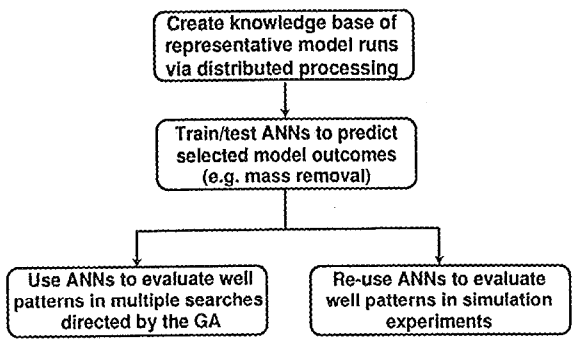


Figure 1. Components of the ANN-GA approach

2. TEST PROBLEM

Figure 2 illustrates the kind of design problem for which the ANN-GA approach was developed. It shows a contour map in parts per billion (ppb) of groundwater contaminated by volatile organic compounds at an industrial site, overlaid by a set of prospective wells intended to clean up the contamination by means of pump-and-treat remediation. In this remedial strategy, contaminated water is pumped to the surface by extraction wells and treated to reduce contamination to acceptable levels. The goal of optimization for this type of problem is usually to find one or more combinations of wells that will at least contain and preferrably clean up the contamination at minimum cost or time. Although the number of well combinations is potentially infinite, it has been customary in groundwater optimization work to prespecify a grid of potentially good well locations and then formulate a combinatorial search to locate the most time- or cost-effective subset of those locations which meets remediation goals.

Triangles mark the location of 30 prospective extraction wells. Their pumping statuses, on vs. off,

are the only design variables in the problem. A well's rate of pumping is tailored to the hydrogeology

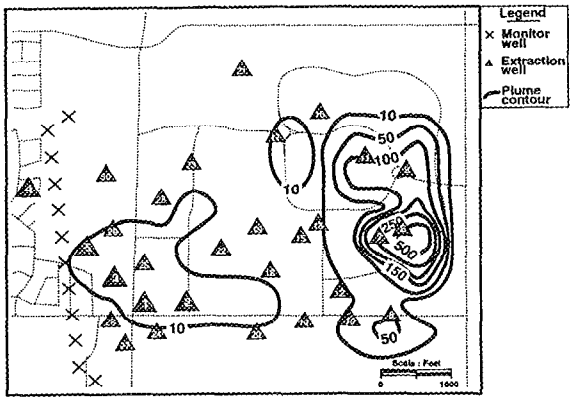


Figure 2. A groundwater remediation problem

of its location, the larger triangles in the figure pumping at 50 gpm while the others pump at 20 gpm. The time-span over which the success or failure of remediation is evaluated is 50 years. The X's to the left of the figure indicate a fence-line of monitoring wells used to determine if the contamination is spreading toward the residential area to the west of the contaminated site.

The first knowledge base for this problem, referred to as Random1, consisted of results from the simulation of 401 well patterns (301 for training plus 100 for testing purposes) randomly sampled from a uniform distribution over the set of 30 possible wells. Both the number of wells in the pattern and the particular wells making up the pattern were randomly selected. The mean number of wells per pattern in the knowledge base was 12.72. The simulator used to predict the effects of pumping was the 2-D hybrid finite-element/finite-difference contaminant transport model SUTRA [4]. Data saved from each simulation included yearly reports of the amount and distribution of contaminant removed, the distribution of residual contamination, and the contamination levels at the key monitoring points. This information could be used to answer questions concerning either cost (e.g. "What is the cheapest way to accomplish remediaion goal X?") or time (e.g. "What is the fastest way to accomplish remediation goal X?").

3. INITIAL RESULTS

The management goal for this problem was to locate the lowest-cost well patterns which prevent the groundwater contamination from spreading beyond the fence-line of monitoring wells. A feed-forward neural network having 30 inputs (one for each of the 30 wells in the problem) and one dichotomous output was trained from 301 patterns in the knowledge base to predict containment according to the following operational definition. The level of contamination predicted by the model at each monitoring well was checked yearly for the entire period of simulation. If the maximum concentration observed at these points remained at or below nine ppb, the pattern was said to contain the contamination. Containment at nine ppb is an easy goal to achieve, as long as cost-control is not an issue. So, 59% of the training patterns met the criterion for containment.

The supervised learning backpropagation algorithm was used in training, employing a conjugate gradient technique to speed convergence and lessen the likelihood of instabilities and oscillations [5]. Parameters used in training included a fixed learning rate of .1 and a sigmoid threshold function. Prior experience had shown that these networks were highly vulnerable to over-fitting; so instead of training to some convergence criterion, training was limited to a fixed number of iterations. Nine variant architectures, differing only in the number of nodes in their hidden layers, were trained and tested. The nine-node variant showed the highest levels of predictive accuracy on the 100 test cases, being 96% correct on the 69 cases that contained the contamination and 100% correct on the 31 cases that failed to contain; consequently, it was selected for use in the search phase. Predictive accuracy is defined throughout this paper as the percentage of agreement between the ANN and SUTRA predictions of containment.

The search was driven by a modified simple GA. The problem representation was the same as that used for the ANN's input layer: a string of 30 locations, each coded as either on or off. Rank-based [6] rather than proportional selection was applied to maintain constant selective pressure throughout the search. Uniform crossover [7] was selected as most appropriate to the problem because physical adjacancy in the problem representation was not related to whether locations would cluster together into effective pumping patterns. Parameters which were held constant throughout all searches were the population size (100), the cross-over probability (.9), the bias used for the rank-order selections (1.5), and the mutation rate (.001). The same 100 patterns used to test the ANNs served as the initial population. The

fitness function was defined as the sum of a pattern's predicted ability to contain the contamination (1=Yes, 0=No) and its installation and maintenance costs over 50 years (rescaled between 0 and 1 and reversed so that the cheaper the pattern, the higher the score). At every generation, the trained network (known as Random1.H9 because it was created from the Random1 knowledge base and contained nine nodes in its hidden layer) was used to generate the containment predictions. A C-function generated the cost information.

Convergence to a single optimal solution is neither practical nor desirable for the kind of design problem being described here. Instead, diversity in the search was encouraged by keeping the crossover rate high, limiting the number of generations to 25, and saving off and analyzing the highest scorers from every generation rather than focusing attention solely on the final generation. So, the solution set that resulted from each search contained at least patterns. Results from three separate searches, differing only in the seed used to initialize the pseudo-random number generator, were combined. After duplicates were eliminated, there were 99 well patterns, ranging in size from three to seven wells and costing from $70 to $130 million, which met the definition of containment as predicted by Random1.H9. (As a point of reference, if all 30 extraction wells were pumped for 50 years, the total cost would be $532 million.) Because cost is heavily affected by pumping rate and wells differ in their rates, the number of wells alone does not dictate the cost; the particular combination of wells is also relevant.

All 99 patterns were submitted to SUTRA for confirmation that they did, in fact, contain the contamination. Despite its high predictive accuracy on the 100 test examples, only 37.4% of the patterns which Random1.H9 had predicted would contain the contamination were confirmed to do so by SUTRA. A breakdown of errors by cost groups indicated that the ANN was particularly inaccurate on the lower-cost well combinations on which the search focuses in its final stages.

4. ALTERNATIVE NETWORKS

To further explore the nature of these predictive errors in the final stages of search and strategies for reducing them, alternative networks were trained and tested following the same procedures as for Random1.H9 and evaluated according to two criteria: 1) how accurately they predicted containment on their own optimal sets (that is, the patterns that were generated in GA searches when the particular alternative network was supplying the predictions of containment), and 2) the absolute number of SUTRA-confirmed very-low-cost patterns resulting from each search.

In all cases, the optimal set was defined as all well patterns costing $130 million or less and predicted by the ANN to contain the contamination. The reason for examining the number of SUTRA-confirmed very-low-cost patterns, is to assess the "yield" of the strategy. For example, an ANN that raises its predictive accuracy by simply avoiding the very-low-cost regions of the search space has limited value compared to a less accurate ANN that still manages to yield a few confirmed very-low-cost patterns. Results are shown in the table below.

ANN	Predictive Accuracy on Optimal Set	Confirmed Patterns <= $100M
Random1.H9	37.4% (N= 99)	13
Random1.H7	79.7% (N= 74)	13
Random1RW.H8	54.9% (N= 82)	16
Random2.H12	51.6% (N=124)	33
Random3.H4	60.0% (N=120)	33

The first alternative network to be evaluated was Random1.H7, the second most accurate variant on the 100 original test examples. This ANN showed far more robust final stage accuracy on the 74 patterns in its own optimal set. Unfortunately, there was no way to know in advance that this particular variant would be a strong performer.

Three *a priori* strategies for increasing a network's resistence to false positives were developed and tested. The first strategy was the least expensive, from a computational standpoint. New training and testing sets, designated as Random1RW, were created out of the original Random1 by making three copies of any well pattern having 13 or fewer wells and only one copy of patterns having more than 13 wells. This trick biased the network's weights in favor of the smaller-sized pattern subregion that would become more important as the search progressed and required

no additional simulations. The second strategy consisted of doubling the number of training examples from 301 to 601 to create the Random2 knowledge base, keeping the distribution of cases over the input parameter space the same as for Random1. This strategy entailed 300 additional model runs. The 100-case testing set remained the same. The third strategy involved doubling the number of training and testing examples involving *13 or fewer wells* and dropping larger patterns out entirely to create the Random3 knowledge base. This strategy keeps the size of the knowledge base (301 training examples, 100 test examples) the same as Random1 but concentrates the examples on the smaller-sized patterns which figure prominantly in cost-minimizing searches.

The particular ANN chosen for evaluation in all three strategies was the hidden-node variant showing the highest accuracy on the appropriate initial test set. In all cases, test set accuracies exceeded 95%. All GA search procedures were the same as described for Random1.H9.

The most effective of the *a priori* strategies, judged both by performance on its own optimal set and its absolute yield, is Random3's focused knowledge base. The reweighting scheme embodied in Random1RW is extremely computationally cheap but less effective by both criteria. The tactic of Random2, simply increasing the number of training cases, has an absolute yield as high as Random3's but less predictive accuracy and is probably not worth the extra simulations it requires. While Random1.H7 is the winner on predictive accuracy, its low yield suggests that it gains its accuracy by avoiding difficult regions of the search space.

5. CONCLUSIONS

Discussions of training factors in supervised learning algorithms tend to take the more-is-better approach to estimating the number of training cases. The results described in this paper suggest that the distribution of cases is as important as absolute numbers. Furthermore, the utility of an ANN must be evaluated not only in terms of test-set generalization but ultimate performance measures such as final-stage accuracy and yield.

6. ACKNOWLEDGEMENTS

This work was supported by Lawrence Livermore National Laboratory's Environmental Restoration Division, under the auspices of the U.S. Department of Energy, contract W-7405-ENG-48.

7. REFERENCES

1 L. L. Rogers and F. U. Dowla (1994). "Optimization of groundwater remediation using artificial neural networks and parallel solute transport modeling," *Water Resources Research,* 30(2), 457-481.

2 L. L. Rogers, F. U. Dowla, and V. M. Johnson (1995). "Optimal field-scale groundwater remediation using neural networks and the genetic algorithm," *Environmental Science & Technology,* 29(5), 1145-1155.

3 V. M. Johnson and L. L. Rogers (1996). "Location analysis in ground-water remediation using neural networks," *Ground Water,* 33(5), 749-758.

4 C. I. Voss (1984). *A finite-element simulation model for saturated-unsaturated, fluid-density-dependent groundwater flow with energy transport or chemically-reactive single-species solute transport.* Washington, D.C.: U. S. Geological Survey, Water Resources Investigations Report #84-4369.

5 E. M. Johansson, F. U. Dowla, and D. M. Goodman (1992). "Backpropagation learning for multi-layer feed-forward neural networks using the conjugate gradient method," *Int. J. of Neural Systems,* 2(4), 291-301.

6 D. Whitley (1989). "The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best," In J. D. Schaffer (Ed.), *Proc. of the Third Int. Conf. on Genetic Algorithms,* San Mateo, CA: Morgan Kaufman, pp. 116-123.

7 G. Syswerda (1989). "Uniform crossover in genetic algorithms," In J. D. Schaffer (Ed.), *Proc. of the Third Int. Conf. on Genetic Algorithms,* San Mateo, CA: Morgan Kaufman, pp. 2-9.